Author: Sinu Sekhar

Date: 08/12/2010

DPAdmin Extensions for Custom XML Modifications

DPAdmin is a highly customizable framework. Shown below is a real customization we did at a customer site. The requirement was to be able to update custom XMLs on the device during the execution of automation scripts. Some of the use cases are shown below.

1. Ability to add, delete, modify routes to a routing XML.
2. Ability to add, delete IPs, hosts etc from IP White list XML.
3. Ability to modify AAA Configuration XMLs by adding, modifying AA configuration that includes usernames, passwords, authorization policy, authentication policies etc.

This document provides a detailed walk through for the Custom Extensions created for DPAdmin at the customer site. Extensions are created in DPAdmin as an ANT task, and invoked using the "custom ant" option both from the Automation Mode as well as the Interactive Mode. Let us take a look at the extensions in detail.

Shown below are the extensions created for our customer.

1. Task for Add/Modify Routing XML.
2. Task for Add/Delete IP White List XML.
3. Task for Add/Delete AAA Info XML with Local users and groups.
4. Task for Add/Delete AAA Info XML with Mutual Auth Subject DN Info.

Task for Add/Modify Routing XML

```xml
<routes>
        <route uri="/service/EmployeeService">
                <protocol>http</protocol>
                <host>127.0.0.1</host>
                <port>8007</port>
        </route>
        <route uri="/service/OrderManagementService">
                <protocol>http</protocol>
                <host>127.0.0.1</host>
                <port>8009</port>
        </route>
```

iSOA Group a Commerce Solutions, Inc. company

San Francisco, CA

www.isoagroup.com

Sample XML is shown above.

This task allows you to modify the routing XML either by adding a new route or by changing an existing route. Although the task itself takes an input file and generates and output file all on the local system, you can combine this with download and upload commands that are native commands to achieve the changes to the XMLs stored on the device directly. We will show you how to do that in a bit.

Let us start with an example. You are already familiar with the way automation commands look in a DPAdmin script. Custom commands are no different.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<dpscript xmlns="http://www.isoagroup.com/datapower/automation/framework/1.0.0.1" name="release07302010-1">
    <dpflow name="deployToNodeXi50-1" environment-id="stg-node1">
        <dpcommand>
            <param name="action" value="include-custom-ant"/>
            <param name="custom-ant-script-file" value="route-task.xml"/>
            <param name="antparam-1" value="action=modify"/>
            <param name="antparam-2" value="input-file=C:/Projects/Visa-DPAdmin/routes.xml"/>
            <param name="antparam-3" value="output-file=C:/Projects/Visa-DPAdmin/routesnew1.xml"/>
            <param name="antparam-4" value="protocol=http"/>
            <param name="antparam-5" value="host=myhost"/>
            <param name="antparam-6" value="port=myport"/>
            <param name="antparam-7" value="route=/websrv_cmservice/v1_0/cardmanagement"/>
        </dpcommand>

        <dpcommand>
            <param name="action" value="include-custom-ant"/>
            <param name="custom-ant-script-file" value="route-task.xml"/>
            <param name="antparam-1" value="action=add"/>
            <param name="antparam-2" value="input-file=C:/Projects/Visa-DPAdmin/routesnew1.xml"/>
            <param name="antparam-3" value="output-file=C:/Projects/Visa-DPAdmin/routesnew2.xml"/>
            <param name="antparam-4" value="protocol=http"/>
            <param name="antparam-5" value="host=myhost"/>
            <param name="antparam-6" value="port=myport"/>
            <param name="antparam-7" value="route=/mynewuri"/>
        </dpcommand>

    </dpflow>

</dpscript>
```

The XML shown above, shows how you can add and modify routes using the custom Route task. Let us look at the properties.

Parameter Name	Parameter Description
action	This parameter contains the various actions that can be used with DPAdmin. In this case, it will always be include-custom-ant, as we are dealing with custom tasks here. Leave it at the default.
custom-ant-script-file	This parameter points to the custom ant file that is leveraged for executing this command. This will also be the same for a particular custom task. ie. all operations to route XML should point to route-task.xml as shown in this example. It will be different for the other custom tasks.
antparam-n	You can see a number of parameters with this name. These parameters should remain the same. This is how DPAdmin accepts parameters to custom ant tasks. The various ANT parameters are explained below.
The ANT parameters are explained in detail. These parameters control the behavior of the custom task and allow you to add/modify route nodes.	
action	This ant parameter determines which action to execute. ie. to add a route node or modify an existing node. The possible name/value combinations are **action=add** and **action=modify.**
input-file	This parameter is used to specify the input file to be used for modification. Folders with spaces are not allowed. Specified as **input-file=<full path with**

	file name>
output-file	This parameter is used to generated output file to be generated with the changes. Folders with spaces are not allowed. Specified as **output-file=<full path with file name>**
protocol	Parameter is used to specify the protocol name to use for the route node. Specified as **protocol=<protocol name>**
host	Parameter is used to specify host name for the route node. Specified as **host=<host name or IP>**
port	Parameter is used to specify port number for the route node. Specified as **port=<port number>**
route	Parameter is used to specify route be added or route to be changed. Specified as **route=<route uri>**

How to combine this task with download and upload commands to make changes to route XML on the device?

```xml
<?xml version="1.0" encoding="UTF-8"?>
<dpscript xmlns="http://www.isoagroup.com/datapower/automation/framework/1.0.0.1" name="rele
        <dpflow name="deployToNodeXi50-1" environment-id="stg-node1">
            <dpcommand>
                <param name="action" value="get-file"/>
                <param name="domain" value="domainname"/>
                <param name="filepath" value="local:///route.xml"/>
                <param name="outfilepath" value="c:/temp/route-original.xml"/>
            </dpcommand>

            <dpcommand>
                <param name="action" value="include-custom-ant"/>
                <param name="custom-ant-script-file" value="route-task.xml"/>
                <param name="antparam-1" value="action=add"/>
                <param name="antparam-2" value="input-file=c:/temp/route-original.xml"/>
                <param name="antparam-3" value="output-file=c:/temp/route-new.xml"/>
                <param name="antparam-4" value="protocol=http"/>
                <param name="antparam-5" value="host=myhost"/>
                <param name="antparam-6" value="port=myport"/>
                <param name="antparam-7" value="route=/mynewuri"/>
            </dpcommand>

            <dpcommand>
                <param name="action" value="copy-file"/>
                <param name="domain" value="domainname"/>
                <param name="source-filename" value="c:/temp/route-new.xml"/>
                <param name="target-filename" value="local:///route.xml"/>
            </dpcommand>
        </dpflow>

</dpscript>
```

As you can see from the example shown above, we have combined the existing API functions to choreograph a powerful operation with a custom task combined. Here are the steps in this script.

1. Download the route XML from DataPower.
2. Use the Custom extension task to modify the XML.
3. Upload the XML back to the device.

You can use this combination of tasks with the other tasks shown below to make changes to the XML directly on the device as well.

Task for Add/Delete IP White List XML

```
<aaa:AAAInfo xmlns:aaa="http://www.datapower.com/AAAInfo">
        <aaa:FormatVersion>1</aaa:FormatVersion>
        <aaa:Filename>local:///ip_whitelist.xml</aaa:Filename>
        <aaa:Summary/><!-- Determine credential from output of the extract-identity phase.-->
        <aaa:Authenticate>
                <aaa:IPHost>192.168.168.1</aaa:IPHost>
                <aaa:OutputCredential>guest</aaa:OutputCredential>
        </aaa:Authenticate>
        <aaa:Authenticate>
                <aaa:IPHost>xxxxxxxxxx</aaa:IPHost>
                <aaa:OutputCredential>guest</aaa:OutputCredential>
        </aaa:Authenticate>
```

Sample XML is shown above.

This task allows you to modify the IP White list XML either by adding a new IP or by deleting an existing IP. Although the task itself takes an input file and generates and output file all on the local system, you can combine this with download and upload commands that are native commands to achieve the changes to the XMLs stored on the device directly as shown above.

Let us start with an example.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<dpscript xmlns="http://www.isoagroup.com/datapower/automation/framework/1.0.0.1" name="release07302010-1">
        <dpflow name="deployToNodeXi50-1" environment-id="stg-node1">
                <dpcommand>
                        <param name="action" value="include-custom-ant"/>
                        <param name="custom-ant-script-file" value="ipwhitelist-task.xml"/>
                        <param name="antparam-1" value="action=add"/>
                        <param name="antparam-2" value="input-file=C:/Projects/Visa-DPAdmin/IP_Whitelist.xml"/>
                        <param name="antparam-3" value="output-file=C:/Projects/Visa-DPAdmin/IP_Whitelist_new.xml"/>
                        <param name="antparam-4" value="ip-address=9.9.9.9"/>
                        <param name="antparam-5" value="credential=admin"/>
                </dpcommand>

        </dpflow>
</dpscript>
```

Parameters are explained below.

Parameter Name	Parameter Description
action	This parameter contains the various actions that can be used with DPAdmin. In this case, it will always be include-custom-ant, as we are dealing with custom tasks here. Leave it at the default.
custom-ant-script-file	This parameter points to the custom ant file that is leveraged for executing this command. This will also be the same for a particular custom task. ie. all operations to IP White List XML should point to ipwhitelist-task.xml as shown in this example. It will be different for the other custom tasks.

antparam-n	You can see a number of parameters with this name. These parameters should remain the same. This is how DPAdmin accepts parameters to custom ant tasks. The various ANT parameters are explained below.
The ANT parameters are explained in detail. These parameters control the behavior of the custom task and allow you to add/modify route nodes.	
action	This ant parameter determines which action to execute. ie. to add an IP or delete an existing IP. The possible name/value combinations are **action=add** and **action=delete.**
input-file	This parameter is used to specify the input file to be used for modification. Folders with spaces are not allowed. Specified as **input-file=<full path with file name>**
output-file	This parameter is used to generated output file to be generated with the changes. Folders with spaces are not allowed. Specified as **output-file=<full path with file name>**
ip-address	Parameter is used to specify the IP Address. Specified as **ip-address=<IP Address>**
credential	Parameter is used to specify the credential for the IP Address. Specified as **credential=<Output Credential>**

Task for Add/Delete AAA Info Node for Local Users

```
<aaa:AAAInfo xmlns:dpfunc="http://www.datapower.com/extensions/functions" xmlns:aaa="http://w
        <aaa:FormatVersion>1</aaa:FormatVersion>
        <aaa:Filename>local:///AAAInfo.xml</aaa:Filename>
        <aaa:Summary/><!-- Determine credential from output of the extract-identity phase. -->
        <aaa:Authenticate>
                <aaa:Username>user</aaa:Username>
                <aaa:Password>password</aaa:Password>
                <aaa:OutputCredential>guest</aaa:OutputCredential>
        </aaa:Authenticate>
        <aaa:Authenticate>
                <aaa:Username>xxxxxx</aaa:Username>
                <aaa:Password>xxxxx</aaa:Password>
                <aaa:OutputCredential>guest</aaa:OutputCredential>
        </aaa:Authenticate>
```

Sample XML is shown above.

This task allows you to modify the AAA Info XML for Local users either by adding a new user or by deleting an existing user. Although the task itself takes an input file and generates and output file all on the local system, you can combine this with download and upload commands that are native commands to achieve the changes to the XMLs stored on the device directly as shown above.

Let us start with an example.

iSOA Group a Commerce Solutions, Inc. company

San Francisco, CA

www.isoagroup.com

```xml
<?xml version="1.0" encoding="UTF-8"?>
<dpscript xmlns="http://www.isoagroup.com/datapower/automation/framework/1.0.0.1" name="release07302010-1">
    <dpflow name="deployToNodeXi50-1" environment-id="stg-node1">
        <dpcommand>
            <param name="action" value="include-custom-ant"/>
            <param name="custom-ant-script-file" value="localuser-aaa-task.xml"/>
            <param name="antparam-1" value="action=add"/>
            <param name="antparam-2" value="input-file=C:/Projects/Visa-DPAdmin/AAA_info.xml"/>
            <param name="antparam-3" value="output-file=C:/Projects/Visa-DPAdmin/AAA_info_new.xml"/>
            <param name="antparam-4" value="username=sinu"/>
            <param name="antparam-5" value="password=password"/>
            <param name="antparam-6" value="credential=admin"/>
        </dpcommand>

    </dpflow>

</dpscript>
```

Parameters are explained below.

Parameter Name	Parameter Description
action	This parameter contains the various actions that can be used with DPAdmin. In this case, it will always be include-custom-ant, as we are dealing with custom tasks here. Leave it at the default.
custom-ant-script-file	This parameter points to the custom ant file that is leveraged for executing this command. This will also be the same for a particular custom task. ie. all operations to AAA Info XML for Local Users should point to localuser-aaa-task.xml as shown in this example. It will be different for the other

	custom tasks.
antparam-n	You can see a number of parameters with this name. These parameters should remain the same. This is how DPAdmin accepts parameters to custom ant tasks. The various ANT parameters are explained below.
The ANT parameters are explained in detail. These parameters control the behavior of the custom task and allow you to add/modify route nodes.	
action	This ant parameter determines which action to execute. ie. to add a local user node or delete an existing user. The possible name/value combinations are **action=add** and **action=delete.**
input-file	This parameter is used to specify the input file to be used for modification. Folders with spaces are not allowed. Specified as **input-file=<full path with file name>**
output-file	This parameter is used to generated output file to be generated with the changes. Folders with spaces are not allowed. Specified as **output-file=<full path with file name>**
username	Parameter is used to specify the username. Specified as **username=<User Name>**
password	Parameter is used to specify the password. Specified as **password=<Password>**
credential	Parameter is used to specify the credential for the User. Specified as **credential=<Output Credential>**

Author: Sinu Sekhar

Date: 08/12/2010

Task for Add/Delete AAA Info Node for Mutual Auth Subject DN

```xml
<?xml version="1.0" encoding="utf-8"?>
<!--
Licensed Materials - Property of IBM
IBM WebSphere DataPower Appliances
Copyright IBM Corporation 2007. All Rights Reserved.
US Government Users Restricted Rights - Use, duplication or disclosure
restricted by GSA ADP Schedule Contract with IBM Corp.
-->
<AAAInfo xmlns="http://www.datapower.com/AAAInfo">
 <FormatVersion>1</FormatVersion>
 <Filename>store:///AAAInfo.xml</Filename>
 <Summary>This is an example of the file format.</Summary>

 <Authenticate>
  <DN>CN=Alice, O=DataPower, L=Cambridge, ST=MA, C=US</DN>
  <OutputCredential>admin</OutputCredential>
 </Authenticate>
 <Authenticate>
  <DN>/C=US/O=isoagroup/OU=gengeme/CN=name</DN>
  <OutputCredential>YTR</OutputCredential>
 </Authenticate>

 <!--Authorize Nodes-->
 <Authorize>
  <InputCredential>YTR</InputCredential>
  <InputResource>/axis2/*</InputResource>
  <Access>allow</Access>
 </Authorize>

</AAAInfo>
```

Sample XML is shown above

This task allows you to modify the AAA Info XML for Mutual Auth Subject DN either by adding a new subject DN or by deleting an existing subject DN as well as Add/Delete Authorize Nodes. Although the task itself takes an input file and generates and output file all on the local system, you can combine this

iSOA Group a Commerce Solutions, Inc. company

San Francisco, CA

www.isoagroup.com

with download and upload commands that are native commands to achieve the changes to the XMLs stored on the device directly as shown above.

Let us start with an example.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<dpscript xmlns="http://www.isoagroup.com/datapower/automation/framework/1.0.0.1" name="release07302010-1">
        <dpflow name="deployToNodeXi50-1" environment-id="stg-node1">
                <dpcommand>
                        <param name="action" value="include-custom-ant"/>
                        <param name="custom-ant-script-file" value="mutualauth-aaa-task.xml"/>
                        <param name="antparam-1" value="action=authenticate-add"/>
                        <param name="antparam-2" value="input-file=C:/Projects/Visa-DPAdmin/mutual_AAA_Info.xml"/>
                        <param name="antparam-3" value="output-file=C:/Projects/Visa-DPAdmin/mutual_AAA_Info_new.xml"/>
                        <param name="antparam-4" value="dn=CN=Alice, O=DataPower, L=Cambridge, ST=MA, C=US"/>
                        <param name="antparam-5" value="credential=admin"/>
                </dpcommand>

                <dpcommand>
                        <param name="action" value="include-custom-ant"/>
                        <param name="custom-ant-script-file" value="mutualauth-aaa-task.xml"/>
                        <param name="antparam-1" value="action=authorize-add"/>
                        <param name="antparam-2" value="input-file=C:/Projects/Visa-DPAdmin/mutual_AAA_Info.xml"/>
                        <param name="antparam-3" value="output-file=C:/Projects/Visa-DPAdmin/mutual_AAA_Info_new.xml"/>
                        <param name="antparam-4" value="resource=/hello/world"/>
                        <param name="antparam-5" value="access=allow"/>
                        <param name="antparam-6" value="credential=admin"/>
                </dpcommand>
        </dpflow>
</dpscript>
```

As shown above, this command not only allows you to add/delete authentication nodes, but also authorize nodes. Let us look at the parameters in detail.

Parameter Name	Parameter Description
action	This parameter contains the various actions that can be used with DPAdmin. In this case, it will always be include-custom-ant, as we are dealing with custom tasks here. Leave it at the default.

custom-ant-script-file	This parameter points to the custom ant file that is leveraged for executing this command. This will also be the same for a particular custom task. ie. all operations to AAA Info XML for Mutual Auth DN should point to mutualauth-aaa-task.xml as shown in this example. It will be different for the other custom tasks.
antparam-n	You can see a number of parameters with this name. These parameters should remain the same. This is how DPAdmin accepts parameters to custom ant tasks. The various ANT parameters are explained below.
The ANT parameters are explained in detail. These parameters control the behavior of the custom task and allow you to add/modify route nodes.	
action	This ant parameter determines which action to execute. ie. to add/delete authentication and authorization nodes. The possible name/value combinations are **action=authenticate-add** and **action=authenticate-delete** for authentication nodes and **action=authorize-add** and **action=authorize-delete** for authorization nodes**.**
input-file	This parameter is used to specify the input file to be used for modification. Folders with spaces are not allowed. Specified as **input-file=<full path with file name>**
output-file	This parameter is used to generated output file to be generated with the changes. Folders with spaces are not allowed. Specified as **output-**

	file=<full path with file name>
dn	Parameter is used to specify the DN. Specified as **dn=<Full DN>**
credential	This parameter serves two purposes. When used with Authenticate, it acts as output credential and when used with Authorize operation it serves as Input Credential. Parameter specified as **credential=<Input or Output Credential based on the type of command>**
resource	Parameter is used to specify the resource for the Authorization. Specified as **resource=<Resource Name>**
access	Parameter is used to specify the access for the Authorization. Specified as **access=<Access Level >**

How to add comment along with the node to four tasks discussed earlier?

Just pass the optional parameter with any of the four tasks mentioned earlier in the format shown below.
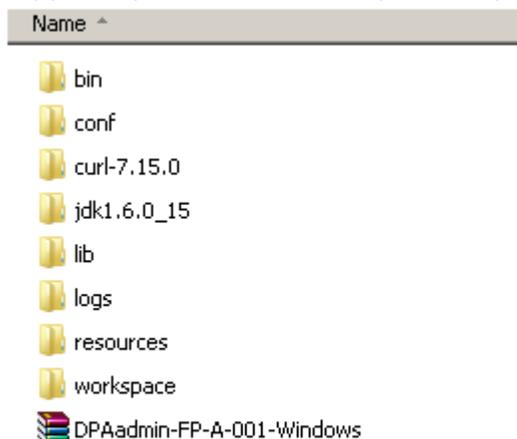
```xml
<dpcommand>
    <param name="action" value="include-custom-ant"/>
    <param name="custom-ant-script-file" value="localuser-aaa-task.xml"/>
    <param name="antparam-1" value="action=add"/>
    <param name="antparam-2" value="input-file=C:/Projects/Visa-DPAdmin/AAA_info.xml"/>
    <param name="antparam-3" value="output-file=C:/Projects/Visa-DPAdmin/AAA_info_new.xml"/>
    <param name="antparam-4" value="username=sinu"/>
    <param name="antparam-5" value="password=password"/>
    <param name="antparam-6" value="credential=admin"/>
    <param name="antparam-7" value="comment=This is a comment"/>
</dpcommand>
```

Please don't forget to increment the antparam number as shown in the example above. This is an optional parameter and can be skipped if not needed. All the other parameters are mandatory.

Fix Pack

All of the custom tasks shown above requires **Fix Pack DPA-A-001** to work. Fix pack can be applied easily to by following the steps below.

1. Stop DPAdmin interactive console or any running automation scripts.
2. Copy the zip file to the location pointed by ISOA_HOME environment variable as shown below.

Name ▲
bin
conf
curl-7.15.0
jdk1.6.0_15
lib
logs
resources
workspace
DPAadmin-FP-A-001-Windows

3. Unzip the file.
4. DPAdmin is ready for use.